

In the Claims:

1. (Currently Amended) A method for writing or otherwise changing data in a non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

(a)[[.]] sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure;

(b)[[.]] sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device;
and

(c)[[.]] sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

2-5. (Canceled)

6. (Original) The method of claim 1, in which ruggedness capability of the device driver can be instructed to be turned on or off.

7. (Currently Amended) A method for writing or otherwise changing data in a unit-based non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

(a)[[.]] sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure;

(b)[[.]] sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device;
and

(c)[[.]] sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

8-11. (Canceled)

12. (Original) The method of claim 7, in which ruggedness capability of the device driver can be instructed to be turned on or off.

13. (Currently Amended) A method for enabling a file system to write or otherwise change data in a non-volatile storage device supported by a ruggedized block device driver, so as to provide ruggedized operation of the file system, the method comprising the steps of:

(a)[[.]] optionally examining each command received by said file system, for determining whether said command should be protected from failures;

and

(b)[[.]] for each said command determined to be protected from failures:

(i)[[.]] sending a command to said device driver for defining current data contents of the storage device as a fall-back state in case of failure;

(ii)[[.]] performing said command received by said file system according to algorithms of said file system, ~~possibly~~ thereby generating at least one command to said device driver, said at least one command potentially changing the data contents of said device; and

(iii)[[.]] sending a command to the device driver for defining said changed data contents of the storage device as a new fall-back state in case of failure.

14. (Original) The method of claim 13, in which said ruggedized operation of said file system can be instructed to be turned on or off.

15. (Currently Amended) A method for converting an existing non-ruggedized file system on a non-volatile storage device supported by a ruggedized block device driver, into a ruggedized file system, the method comprising the steps of:

(a) adding, in the beginning of the file system code implementing each file system command which might change data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure; and

(b) adding, at the end of the file system code implementing each file system command which might change said data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure;

wherein, for each said file system command which might change said data contents of the storage device, if a failure occurs after said command to the device driver of step (a) is sent to the device driver but before said sending to the device driver of said command to the device driver of step (b) is completed, the device driver rolls back the effects of said each file system command which might change data contents of the storage device and returns the storage device to contain said data contents defined as a fall-back state by said command to the device driver of step (a);

and wherein, for each said file system command which might change said data contents of the storage device, the device driver identifies data associated with said each file system command which might change said data contents of the storage device, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is

stored in said physical blocks that are not the first blocks in said chains of physical blocks.

16. (Original) The method of claim 15 in which the resulting ruggedized file system is compatible with said existing not ruggedized file system on the same physical device, such that either a physical device operated under said ruggedized file system can be operated under said existing file system, or a physical device operated under said existing file system can be operated under said ruggedized file system, without loss of data when changing between one file system and the other file system.

17. (Currently Amended) A method for a software application to write or otherwise change data on a non-volatile storage device, where the storage device is supported by a ruggedized block device driver and a file system, so as to provide ruggedized operation of the application, the method comprising the steps of:

(a)[[.]] sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure;

(b)[[.]] sending a sequence of at least one command to the file system, each said command potentially changing said data contents of the device;
and

(c)[[.]] sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a);

and wherein the device driver identifies data associated with said at least one command conducted after establishing said fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

18. (Currently Amended) A method for converting an existing non-ruggedized application using a non-volatile storage device supported by a ruggedized block device driver and a file system, into a ruggedized application, the method comprising the steps of:

- (a)[[.]] adding, before code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure; and
- (b)[[.]] adding, after said code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure;

wherein, for each said code sending any sequence of commands to the file system which might change the file system's data contents, if a failure occurs after said command to the device driver of step (a) is sent to the device driver but before said sending to the device driver of said command to the device driver of step (b) is completed, the device driver rolls back the effects of said each code sending any sequence of commands to the file system which might change the file system's data

contents and returns the storage device to contain said data contents defined as a fall-back state by said command to the device driver of step (a);
and wherein, for each said code sending any sequence of commands to the file system which might change the file system's data contents, the device driver identifies data associated with said each code sending any sequence of commands to the file system which might change the file system's data contents, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

19. (Currently Amended) A system providing ruggedized operation of a non-volatile storage device, comprising:

(a)[[.]] physical non-volatile storage media; and

(b)[[.]] a software block device driver which is ruggedized by itself,
independently of a file system or other software application using it;

wherein said device driver identifies data associated with new operations conducted after establishing a fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, such that all such new data is stored in said blocks which are not the first blocks in said chains.

20. (Original) The system of claim 19, in which said physical non-volatile storage media is unit-based media.

21-23. (Canceled)

24. (Original) The system of claim 19, in which ruggedness capability of said ruggedized device driver can be instructed to be turned on or off.

25. (Currently Amended) A system providing ruggedized operation of a file system on a non-volatile storage device, comprising the following:

(a)[[.]] physical non-volatile storage media;

(b)[[.]] a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it; and

(c)[[.]] a ruggedized file system wherein ruggedness of said file system is achieved by using the ruggedized features of said block device driver;

wherein said device driver identifies data associated with new operations conducted after establishing a fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, such that all such new data is stored in said blocks which are not the first blocks in said chains.

26. (Original) The system of claim 25, wherein said ruggedized file system is compatible with at least one non-ruggedized file system on the same physical device, such that either a physical device operated under said ruggedized file system can be operated under said non-ruggedized file system, or a physical device operated under said non-ruggedized file system can be operated under said ruggedized file system, without loss of data when changing from one file system to the other file system.

27. (Original) The system of claim 25, in which ruggedness capability of said ruggedized file system can be instructed to be turned on or off.

28. (Currently Amended) A system providing ruggedized operation of a software application on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it;
- (c) a file system; and
- (d) a software application, such that ruggedness of said application is achieved by using ruggedized features of said block device driver;

wherein said device driver identifies data associated with new operations conducted after establishing a fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, such that all such new data is stored in said blocks which are not the first blocks in said chains.

29. (New) A method for writing or otherwise changing data in a non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

- (a) sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure;
 - (b) sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device;
- and

- (c) sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes..

30. (New) A method for writing or otherwise changing data in a unit-based non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

- (a) sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure;
- (b) sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device;
and
- (c) sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes..

31. (New) A method for converting an existing non-ruggedized file system on a non-volatile storage device supported by a ruggedized block device driver, into a ruggedized file system, the method comprising the steps of:

- (a) adding, in the beginning of the file system code implementing each file system command which might change data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure; and
- (b) adding, at the end of the file system code implementing each file system command which might change said data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure;

wherein, for each said file system command which might change said data contents of the storage device, if a failure occurs after said command to the device driver of step (a) is sent to the device driver but before said sending to the device driver of said

command to the device driver of step (b) is completed, the device driver rolls back the effects of said each file system command which might change data contents of the storage device and returns the storage device to contain said data contents defined as a fall-back state by said command to the device driver of step (a);

and wherein, for each said file system command which might change said data contents of the storage device, the device driver identifies data associated with said each file system command which might change said data contents of the storage device, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes.

32. (New) A method for a software application to write or otherwise change data on a non-volatile storage device, where the storage device is supported by a ruggedized block device driver and a file system, so as to provide ruggedized operation of the application, the method comprising the steps of:

- (a) sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure;
- (b) sending a sequence of at least one command to the file system, each said command potentially changing said data contents of the device;
and
- (c) sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said at least one command conducted after establishing said fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes.

33. (New) A method for converting an existing non-ruggedized application using a non-volatile storage device supported by a ruggedized block device driver and a file system, into a ruggedized application, the method comprising the steps of:

- (a) adding, before code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure; and
- (b) adding, after said code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure;

wherein, for each said code sending any sequence of commands to the file system which might change the file system's data contents, if a failure occurs after said command to the device driver of step (a) is sent to the device driver but before said sending to the device driver of said command to the device driver of step (b) is

completed, the device driver rolls back the effects of said each code sending any sequence of commands to the file system which might change the file system's data contents and returns the storage device to contain said data contents defined as a fall-back state by said command to the device driver of step (a);

and wherein, for each said code sending any sequence of commands to the file system which might change the file system's data contents, the device driver identifies data associated with said each code sending any sequence of commands to the file system which might change the file system's data contents, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes..

34. (New) A system providing ruggedized operation of a non-volatile storage device, comprising:

- (a) physical non-volatile storage media; and
- (b) a software block device driver which is ruggedized by itself, independently of a file system or other software application using it;

wherein said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along the chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after said points of change.

35. (New) A system providing ruggedized operation of a file system on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it; and
- (c) a ruggedized file system wherein ruggedness of said file system is achieved by using the ruggedized features of said block device driver;

wherein said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along the chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after said points of change.

36. (New) A system providing ruggedized operation of a software application on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it;
- (c) a file system; and
- (d) a software application, such that ruggedness of said application is achieved by using ruggedized features of said block device driver

wherein said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along

the chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after said points of change.

37. (New) A method for writing or otherwise changing data in a non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

- (a) sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure;
- (b) sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device;
and
- (c) sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

38. (New) A method for writing or otherwise changing data in a unit-based non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

- (a) sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure;
- (b) sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device; and
- (c) sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

39. (New) A method for converting an existing non-ruggedized file system on a non-volatile storage device supported by a ruggedized block device driver, into a ruggedized file system, the method comprising the steps of:

- (a) adding, in the beginning of the file system code implementing each file system command which might change data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure; and

- (b) adding, at the end of the file system code implementing each file system command which might change said data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure;

wherein, for each said file system command which might change said data contents of the storage device, if a failure occurs after said command to the device driver of step (a) is sent to the device driver but before said sending to the device driver of said command to the device driver of step (b) is completed, the device driver rolls back the effects of said each file system command which might change data contents of the storage device and returns the storage device to contain said data contents defined as a fall-back state by said command to the device driver of step (a)

and wherein, for each said file system command which might change said data contents of the storage device, the device driver identifies data associated with said each file system command which might change said data contents of the storage device, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

40. (New) A method for a software application to write or otherwise change data on a non-volatile storage device, where the storage device is supported by a ruggedized block device driver and a file system, so as to provide ruggedized operation of the application, the method comprising the steps of:

- (a) sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure;

- (b) sending a sequence of at least one command to the file system, each said command potentially changing said data contents of the device;
and
- (c) sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure;

wherein if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a); and wherein the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

41. (New) A method for converting an existing non-ruggedized application using a non-volatile storage device supported by a ruggedized block device driver and a file system, into a ruggedized application, the method comprising the steps of:

- (a) adding, before code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure; and
- (b) adding, after said code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure;

wherein, for each said code sending any sequence of commands to the file system which might change the file system's data contents, if a failure occurs after said command to the device driver of step (a) is sent to the device driver but before said sending to the device driver of said command to the device driver of step (b) is completed, the device driver rolls back the effects of said each code sending any sequence of commands to the file system which might change the file system's data contents and returns the storage device to contain said data contents defined as a fall-back state by said command to the device driver of step (a);

and wherein, for each said code sending any sequence of commands to the file system which might change the file system's data contents, the device driver identifies data associated with said each code sending any sequence of commands to the file system which might change the file system's data contents, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

42. (New) A system providing ruggedized operation of a non-volatile storage device, comprising:

- (a) physical non-volatile storage media; and
- (b) a software block device driver which is ruggedized by itself, independently of a file system or other software application using it;

wherein said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals the global generation value.

43. (New) A system providing ruggedized operation of a file system on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it; and
- (c) a ruggedized file system wherein ruggedness of said file system is achieved by using the ruggedized features of said block device driver.

wherein said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals the global generation value.

44. (New) A system providing ruggedized operation of a software application on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it;
- (c) a file system; and
- (d) a software application, such that ruggedness of said application is achieved by using ruggedized features of said block device driver.

wherein said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a generation field with

each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals the global generation value.

45. (New) A method for writing or otherwise changing data in a non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

- (a) adding, to the device driver, only:
 - (i) a first command for defining current data contents of the storage device as a fall-back state in case of failure, and
 - (ii) a second command for defining current data contents of the storage device as a new consistent state;
- (b) sending said first command to the device driver;
- (c) sending to the device driver a sequence of one or more commands that potentially change the data contents of the device; and
- (d) sending said second command to the device driver.

46. (New) The method of claim 45, wherein if a failure occurs after step (b) but before the completion of step (d), the device driver rolls back the effects of all said commands issued in step (c) and returns the storage device to contain said data contents defined as a fall-back state in step (b).

47. (New) The method of claim 46 in which the device driver identifies data associated with said commands conducted after establishing said fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks,

and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

48. (New) The method of claim 46 in which the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes.

49. (New) The method of claim 46 in which the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

50. (New) The method of claim 45, in which ruggedness capability of the device driver can be instructed to be turned on or off.

51. (New) A method for writing or otherwise changing data in a unit-based non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of:

- (a) adding, to the device driver, only:
 - (i) a first command for defining current data contents of the storage device as a fall-back state in case of failure, and

- (ii) a second command for defining current data contents of the storage device as a new consistent state;
- (b) sending said first command to the device driver;
- (c) sending to the device driver a sequence of one or more commands that potentially change the data contents of the device; and
- (d) sending said second command to the device driver.

52. (New) The method of claim 51, wherein if a failure occurs after step (b) but before the completion of step (d), the device driver rolls back the effects of all said commands issued in step (c) and returns the storage device to contain said data contents defined as a fall-back state in step (b).

53. (New) The method of claim 52 in which the device driver identifies data associated with said commands conducted after establishing said fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

54. (New) The method of claim 52 in which the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes.

55. (New) The method of claim 52 in which the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

56. (New) The method of claim 51, in which ruggedness capability of the device driver can be instructed to be turned on or off.

57. (New) A method for enabling a file system to write or otherwise change data in a non-volatile storage device supported by a block device driver, so as to provide ruggedized operation of the file system, the method comprising the steps of:

- (a) adding, to the device driver, only:
 - (i) a first command for defining current data contents of the storage device as a fall-back state in case of failure, and
 - (ii) a second command for defining current data contents of the storage device as a new consistent state;
- (b) optionally examining each command received by the file system, for determining whether said command should be protected from failures; and
- (c) for each said command determined to be protected from failures:
 - (i) sending said first command to said device driver, by the file system;

- (ii) performing said command received by said file system according to algorithms of said file system, thereby generating a sequence of at least one command to said device driver that potentially changes the data content of the device; and
- (iii) sending said second command to the device driver, by the file system.

58. (New) The method of claim 57, in which said ruggedized operation of said file system can be instructed to be turned on or off.

59. (New) A method for converting an existing non-ruggedized file system on a non-volatile storage device supported by a block device driver, into a ruggedized file system, the method comprising the steps of:

- (a) adding, to the the device driver, only:
 - (i) a first command for defining current data contents of the storage device as a fall-back state in case of failure, and
 - (ii) a second command for defining current data contents of the storage device as a new consistent state;
- (b) adding, in the beginning of the file system code implementing each file system command which might change data contents of the storage device, new code for sending said first command to the device driver; and
- (c) adding, at the end of the file system code implementing each file system command which might change said data contents of the storage

device, new code for sending said second command to the device driver.

60. (New) The method of claim 59 in which the resulting ruggedized file system is compatible with said existing not ruggedized file system on the same physical device, such that either a physical device operated under said ruggedized file system can be operated under said existing file system, or a physical device operated under said existing file system can be operated under said ruggedized file system, without loss of data when changing between one file system and the other file system.

61. (New) A method for a software application to write or otherwise change data on a non-volatile storage device, where the storage device is supported by a block device driver and a file system, so as to provide ruggedized operation of the application, the method comprising the steps of:

- (a) adding, to the the device driver, only:
 - (i) a first command for defining current data contents of the storage device as a fall-back state in case of failure, and
 - (ii) a second command for defining current data contents of the storage device as a new consistent state;
- (b) sending, by the software application, said first command to the device driver;
- (c) sending, by the software application, a sequence of at least one command to the file system, each said command potentially changing said data contents of the device; and

- (d) sending, by the software application, said second command to the device driver.

62. (New) A method for converting an existing non-ruggedized application using a non-volatile storage device supported by a block device driver and a file system, into a ruggedized application, the method comprising the steps of:

- (a) adding, to the the device driver, only:
 - (i) a first command for defining current data contents of the storage device as a fall-back state in case of failure, and
 - (ii) a second command for defining current data contents of the storage device as a new consistent state;
- (b) adding, before code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending said first command to the device driver; and
- (c) adding, after said code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending said second command to the device driver.

63. (New) A system providing ruggedized operation of a non-volatile storage device, comprising:

- (a) physical non-volatile storage media; and
- (b) a software block device driver which is ruggedized by itself, independently of a file system or other software application using it, by supporting a set of commands that includes only:

- (i) a first command for defining current data contents of the storage media as a fall-back state in case of failure,
- (ii) a second command for defining current data contents of the storage media as a new consistent state, and
- (iii) commands supported by a corresponding non-ruggedized software block device driver.

64. (New) The system of claim 63, in which said physical non-volatile storage media is unit-based media.

65. (New) The system of claim 63, in which said device driver identifies data associated with new operations conducted after establishing a fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, such that all such new data is stored in said blocks which are not the first blocks in said chains.

66. (New) The system of claim 63, in which said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along the chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after said points of change.

67. The system of claim 63, in which said device driver identifies the data associated with new operations conducted after establishing a fall-back state, by

associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals the global generation value.

68. (New) The system of claim 63, in which ruggedness capability of said ruggedized device driver can be instructed to be turned on or off.

69. (New) A system providing ruggedized operation of a file system on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it, by supporting a set of commands that includes only:
 - (i) a first command for defining current data contents of the storage media as a fall-back state in case of failure,
 - (ii) a second command for defining current data contents of the storage media as a new consistent state, and
 - (iii) commands supported by a corresponding non-ruggedized software block device driver; and
- (c) a ruggedized file system wherein ruggedness of said file system is achieved by using the ruggedized features of said block device driver.

70. (New) The system of claim 69, wherein said ruggedized file system is compatible with at least one non-ruggedized file system on the same physical device,

such that either a physical device operated under said ruggedized file system can be operated under said non-ruggedized file system, or a physical device operated under said non-ruggedized file system can be operated under said ruggedized file system, without loss of data when changing from one file system to the other file system.

71. (Original) The system of claim 69, in which ruggedness capability of said ruggedized file system can be instructed to be turned on or off.

72. (New) A system providing ruggedized operation of a software application on a non-volatile storage device, comprising the following:

- (a) physical non-volatile storage media;
- (b) a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it by supporting a set of commands that includes only:
 - (i) a first command for defining current data contents of the storage media as a fall-back state in case of failure,
 - (ii) a second command for defining current data contents of the storage media as a new consistent state, and
 - (iii) commands supported by a corresponding non-ruggedized software block device driver;
- (c) a file system; and
- (d) a software application, such that ruggedness of said application is achieved by using ruggedized features of said block device driver.